

Projektprotokoll – Monster Trading Cards Game (MTCG)

Projektname: Monster Trading Cards Game (MTCG)

Erstellungsdatum: 27.10.2024

GitHub Repo Link: <https://github.com/Specifik/MTCG>

1. Einleitung

Das Projekt „Monster Trading Cards Game (MTCG)“ ist eine serverseitige Anwendung, die Benutzerverwaltung und Spielsteuerung implementiert. Es basiert auf einer serviceorientierten Architektur mit einer klaren Trennung zwischen Service-, Controller- und Datenzugriffsschicht, um die Wartbarkeit und Erweiterbarkeit der Anwendung zu gewährleisten.

2. Designentscheidungen und Architektur

Die Anwendung verwendet eine REST-Architektur mit dem Repository- und Unit-of-Work-Muster, um eine konsistente und sichere Datenverwaltung zu gewährleisten. Die Passwortsicherheit wird durch SHA-256-Hashing sichergestellt. Hauptmodule und ihre Rollen:

- **UserService:** Vermittelt zwischen HTTP-Anfragen und der Business-Logik.
- **UserController:** Führt spezifische Benutzeroperationen aus und verwaltet Benutzeraktionen.
- **UserRepository:** Bietet eine Abstraktionsschicht für den Datenzugriff, die direkte Datenbankoperationen von der Geschäftslogik trennt.
- **PasswordHasher:** Unterstützt die Passwortsicherheit durch Hashing.

3. Klassenstruktur und -beschreibung

3.1 UserService

UserService implementiert das Service-Interface und verarbeitet HTTP-Anfragen, indem es Anfragen wie Benutzerregistrierungen oder -abfragen an den UserController weiterleitet.

- Methode `handleRequest(Request request)`: Diese Methode verarbeitet GET- und POST-Anfragen und ruft die entsprechenden Controller-Methoden auf.

3.2 UserController

UserController implementiert die Geschäftslogik für Benutzerverwaltung und Authentifizierung. Er verwendet UserRepository für Datenbankoperationen und PasswordHasher für die Passwortverschlüsselung.

3.3 UserRepository

Die Klasse UserRepository ist die zentrale Schnittstelle für den Datenbankzugriff in der Benutzerverwaltung. Sie kapselt die Datenbanklogik und ermöglicht so eine klare Trennung zwischen Datenzugriff und Geschäftslogik.

3.4 PasswordHasher

PasswordHasher ist eine Utility-Klasse, die Passwörter mithilfe des SHA-256-Algorithmus verschlüsselt, um eine sichere Speicherung zu gewährleisten.

4. Klassendiagramm (UML)

Projekt Klassendiagramm:

